THE OFFICE OF THE CHIEF INFORMATION OFFICER
ENTERPRISE TECHNOLOGY STRATEGIES

North Carolina Statewide Technical Architecture

Domain White Paper

Platform Architecture Technology Overview

STATEWIDE TECHNICAL ARCHITECTURE

Domain White Paper: Platform Architecture Technology Overview

Initial Released Date:	August 1, 2003	Version:	1.0.0
Revision Approved Date:	Not Applicable		
Date of Last Review:	March 11, 2004	Version:	1.0.1
Date Retired:			
Architecture Interdependencies:			
Reviewer Notes: Reviewed and updated office title and copyright date. Added a hyperlink for the ETS email – March 11, 2004.			

Office of the Chief Information Officer
Enterprise Technology Strategies
PO Box 17209
Raleigh, North Carolina 27699-7209
Telephone (919) 981-5510
ets@ncmail.net

Mission Statement

Platform Architecture identifies hardware platforms and associated operating systems supporting the state's business.

North Carolina, along with most private enterprises, is supplementing or replacing an inventory of traditional, dumb terminal application portfolios with distributed, personal computer-fronted or Web, graphical applications. End users have grown accustomed to graphical user interfaces (GUIs) and prefer them over the character interfaces of dumb terminals. The graphical user interface has become the centerpiece of the user's environment. Additional interfaces such as voice, video, smart cards and others have extended the options available to address specific business requirements and reach a wider user base.

The change in focus to a graphical interface has implications for application architecture and platform selection. In the past, the central processor and its associated software determined the boundaries for a user. Today, distributed computing client/server models focus on bringing a variety of resources to make end users more productive. A variety of platforms provide the services that meet end-user needs.

The platform decision for end users, applications or databases is dependent on the overall business requirements. Platform Architecture is a physical implementation of a logical application design. Adopting an N-tier client/server application architecture requires that the software tiers are implemented on hardware tiers. Hardware tiers imply different kinds of computers performing different functions. In a three-tier hardware configuration, the tiers often comprise 1.) PC clients, 2.) local servers and 3.) one or more remote servers.

Note: Additional information about Thin Client, Storage, and Mobile Computing will be added to this Platform Domain in future releases.

Three software tiers may execute on a single platform or multiple platforms from a hardware viewpoint. For example, with a mainframe application, the user interface, business logic, and data access all reside on one platform. In a distributed client/server environment, the user interface may run on a PC, local servers may process business logic, and data may reside on the mainframe. A properly designed application facilitates changes in platform strategy with minimal impact on operations. For example, it should be possible to move the database to a different

platform without modifying the application architecture, or change the user interface with minimal or no modifications to application software.

Hardware tiers exist to maximize the usefulness of various specialized hardware devices. Specific platform choices reflect the relative cost, reliability, speed, I/O throughput, disk capacity and graphical capabilities of the different platforms. For example, PCs are good platforms for presentation services because of the low hardware cost and readily accessible graphical I/O capabilities. Mainframes and midrange UNIX and Windows NT platforms are good application servers or database servers.

In an N-tiered, distributed client/server environment, the mainframe becomes just another server. Through the use of integration strategies (see the System Integration Architecture Domain) the life of monolithic mainframe legacy applications can be extended.

Moving to an N-tiered, distributed client/server application architecture introduces platform and operating system infrastructure and integration challenges. Interoperability is defined as "devices and services that can be connected and work together to exchange information among multiple, diverse systems so that information can be processed meaningfully". Interoperability is a key requirement across all platforms. Adherence to the platform architecture technology component principles and standards outlined in this chapter will facilitate interoperability across the state's diverse existing and new platforms. It will also provide consistent platforms that enable easier integration.

The Platform Architecture describes the platform requirements for building a client/server infrastructure. Technical topics to be discussed under platform architecture include client architecture and server architecture.

Client Architecture the client side of client/server is the interface to the application. Usually, a client requires an operating system. Advances in technology and connectivity have increased the types of client interfaces. These clients include smart cards, PDAs, pen computers, voice response units, optical devices and others.

Note: Additional information about client side architecture, including mobile computing, will be addressed in the next release of this Domain.

Server Architecture The server provides services requested by clients. The server application runs on top of an operating system. Servers are often specialized by the type of function they perform on behalf of the client. They include application servers, database servers, and file and print servers.

Middleware This is software that runs on both the client and server side of the application. Ensuring that application middleware accommodates your selection of client and server platforms is important. (Application Communication Middleware is discussed in the System Integration Domain within the Statewide Technical Architecture.)